



## Historia de SQL

### Transcripción

[00:00] Y bueno, ahora vamos a hablar un poco sobre la historia de SQL. Se puede pronunciar SQL, SQL, pero vamos a pronunciarlo SQL, vamos a acostumbrarnos que nos queda más fácil de esa forma. Hay una controversia muy grande en cómo pronunciarlo correctamente, pero nosotros simplemente vamos a hablar de SQL aquí.

[00:21] Entonces, ya que ustedes están comenzando con este lenguaje, es importante que conozcan un poco de su historia, por qué surgió, cuáles fueron los motivos que generaron la necesidad de desarrollar un lenguaje así. Entonces veamos aquí un poco de la historia. En los años 70, en los laboratorios de IBM en San José, California, se estaba realizando una investigación.

[00:45] Y se quería observar la viabilidad de la implementación de un sistema de consultas a bases de datos relacionales ideado por el británico Edgar Frank Codd. Él ideó una forma sencilla de manipular y extraer no solamente los datos, sino de las estructuras de los datos, aprovechando este modelo relacional de bases de datos, conocido como DBMS: Data Base Management Systems.

[01:21] Antiguamente, las bases de datos eran secuenciales, eran una tira de caracteres de texto que la única forma de accederlos era siguiéndola, entonces no había una forma de hallar una relación entre ellos, lo que impedía que se facilitase, el hallar estos datos de una forma como más específica y con el

modelo propuesto por Codd, entonces se vio una salida o se vio mejor una alternativa para poder trabajar con estas bases de datos relacionales.

[01:56] En los años 80 surgieron otras empresas como Oracle, como SQL Server de Microsoft, que comenzaron también a utilizar este lenguaje SQL para trabajar con bases de datos, desde luego, y al ver este surgimiento de tantas empresas queriendo trabajar con SQL, entonces entró el Instituto Americano Nacional de Estándares o la ANSI a crear una regulación para este lenguaje.

[02:27] Dijeron: "Vamos a establecer unos estándares", algo análogo a lo que hizo el modelo OSI de telecomunicaciones. Hagamos un paréntesis aquí, que ellos crearon un modelo de intercomunicación en los equipos, a través de capas, etcétera, pero aquí estamos hablando de SQL, entonces la ANSI ellos establecieron una serie de parámetros para que este lenguaje funcionara respetando unas normas.

[02:58] Y allí surgió entonces SQL, el lenguaje SQL, que funciona siguiendo o respetando CRUD, que es Create Read Update Delete, o sea crea, lee, actualiza y borra. Eso es como un acrónimo que nos ayuda mucho a recordar los principios básicos de SQL.

[03:24] Entonces surgieron, la ANSI dividió estos comandos en tres grandes grupos. Entonces serían DDL, que es Data Definition Language, que es el que se encarga de manipular todas las estructuras de las bases de datos en sí. Entonces, por ejemplo, allí se usa el comando create para crear una base de datos o una tabla.

[03:50] Create table o drop, que sería para remover una tabla, remover una base de datos, estos comandos los estaremos viendo en el desarrollo del curso. También están los DML que es Data Manipulation Lenguaje o el lenguaje de manipulación de los datos, que ya sería select, para seleccionar unos datos, insert para insertar datos a las tablas.

[04:16] También update, que sería actualizar estos datos o delete, que sería borrarlos y Data Control Language, que ya es el que se encarga de la administración en sí de toda la estructura de la base de datos. Entonces, por ejemplo, está la administración de la base de datos, la administración de los accesos a usuarios de los logs, es toda la parte de políticas de crecimiento de la base de datos, etcétera.

[04:46] Entonces esto es como a grandes rasgos lo que hizo la ANSI estableciendo unos estándares. Entonces ello pues generó que surgieran, pues empresas, Oracle, MySQL, SQL Server, entre otras, respetando estos parámetros establecidos por la ANSI. Entonces SQL tiene las siguientes ventajas, las principales: Costo reducido de aprendizaje.

[05:14] Claro, porque cuando una empresa o cuando una persona o una organización trabaja con SQL, entonces si trabaja con Oracle o trabaja con MySQL fácilmente pues pueden aprender el otro lenguaje, porque en principio ANSI está siendo respetado, que es el mismo para ellas.

[05:37] O SQL Server es fácil de aprender, porque ya estos principios están allí como de cierta forma integrados. La portabilidad. Si eventualmente la empresa quiere migrar de base de datos, entonces no está restringida, sino que como trabaja con SQL entonces fácilmente puede ir de Oracle a SQL Server, de SQL Server a MySQL, de MySQL a Postgre.

[06:05] La longevidad. Va a durar por más tiempo, porque como también hablan el mismo lenguaje estandarizado, entonces cuando surjan nuevas actualizaciones, pues no se va a caer la base de datos, sino que va a permanecer activa.

[06:22] Comunicación. Bueno, lo mismo, habrá facilidad de integración en procesos, de diferentes bases de datos, procedimientos de TL o integración, como lo mencioné previamente. Todo esto se facilita porque están hablando el mismo lenguaje SQL.

[06:41] Y libertad de elección. Entonces, ya las empresas no se tienen que preocupar si este lenguaje va a ser compatible con el otro, porque el lenguaje nativo es el mismo, digamos ellas comparten el mismo lenguaje nativo establecido por la ANSI, de modo que no hay ningún problema en ello de ir de Oracle y comenzará a utilizar MySQL Server, por ejemplo.

[07:05] De la misma forma, la persona o las empresas se pueden concentrar entonces en otros aspectos a la hora de escoger su base de datos, como rendimiento o desempeño, hardware, entre otras características. Esas son las principales ventajas, pero desde luego que tiene sus desventajas.

[07:23] Entonces, hoy en día con la creciente demanda de las redes sociales, del Big Data, entonces faltó creatividad, o sea, cuando se creó ese SQL pues no se consideró tanto el impacto que tendría décadas después, como es el caso de hoy en día, entonces esto limita mucho el lenguaje.

[07:45] Porque es muchos datos de muchísima información y encajarla en este patrón estructurado de bases de datos relacionales, entonces es difícil. Otra desventaja, pues, es que surgió NoSQL. El surgimiento de lenguajes que no siguen los estándares ANSI, entonces no son SQL. [08:11] O sea es un lenguaje que ya sí puede atender mejor la demanda de la actualidad de los datos, pero ya es más específico, porque es que digamos hoy en día, con tantos datos, lógicamente ya tenerlos en una estructura en sí es muy complicado. Entonces NoSQL ya resuelve ese problema, pero el problema de NoSQL es que ya es más específico.

[08:37] Entonces NoSQL trabaja de forma como más específica, más puntual, con los datos, y no es fácil entre diferentes empresas tener como es reducir el costo de aprendizaje porque pues pueden saber un lenguaje NoSQL pero no conocen otro, por ejemplo. Otra desventaja es la falta de más estructuración a su lenguaje.

[09:02] Como les digo, todos se ciñeron a la ANSI, pero de cierta forma lo que diferencia a MySQL de Oracle, de Postgre, de SQL Server de Microsoft, lo que los diferencia es que cada una de ellas, usando, partiendo o sea de la base que tiene el mismo lenguaje nativo, también crearon sus propios lenguajes convencionales de programación al interior de SQL.

[09:27] Entonces colocaron por ejemplo, MySQL tiene ifs, for, etcétera, que son ya comandos que permiten trabajar un poco mejor con SQL. Entonces esto es a grandes rasgos, a muy grandes rasgos, la historia de SQL. Hablamos un poco sobre la ANSI también y principales ventajas y desventajas. El próximo video estaremos hablando entonces sobre la historia de MySQL.